

ARTIFICIAL NEURAL NETWORK APPROACH FOR THE IDENTIFICATION OF CLOVE BUDS ORIGIN BASED ON METABOLITES COMPOSITION

RUSTAM^{a,*}, AGUS YODI GUNAWAN^a, MADE TRI ARI PENIA KRESNOWATI^b

^a *Institut Teknologi Bandung, Faculty of Mathematics and Natural Sciences, Industrial and Financial Mathematics Research Group, Jl. Ganesha 10, 40132 Bandung, Indonesia*

^b *Institut Teknologi Bandung, Faculty of Industrial Technology, Food and Biomass Processing Technology Research Group, Jl. Ganesha 10, 40132 Bandung, Indonesia*

* corresponding author: rustam.math@itb.ac.id

ABSTRACT. This paper examines the use of an artificial neural network approach in identifying the origin of clove buds based on metabolites composition. Generally, large data sets are critical for an accurate identification. Machine learning with large data sets lead to a precise identification based on origins. However, clove buds uses small data sets due to the lack of metabolites composition and their high cost of extraction. The results show that backpropagation and resilient propagation with one and two hidden layers identifies the clove buds origin accurately. The backpropagation with one hidden layer offers 99.91 % and 99.47 % for training and testing data sets, respectively. The resilient propagation with two hidden layers offers 99.96 % and 97.89 % accuracy for training and testing data sets, respectively.

KEYWORDS: Artificial neural networks, backpropagation, resilient propagation, clove buds.

1. INTRODUCTION

There is a variation in the flavour and aroma of different plantation commodities. For example, in Indonesia, the clove buds from Java have a prominent wooden aroma and sour flavour while those in Bali have a sweet-spicy flavour [1]. Arabica coffee from Gayo has a lower acidity and a strong bitterness. In contrast, coffee from Toraja has a medium browning, tobacco, or caramel flavour, not too acidic and bitter. Furthermore, Kintamani coffee from Bali has a fruit flavour and acidity, mixed with a fresh flavour. Contrastingly, Coffee from Flores has a variety of flavours ranging from chocolate, spicy, tobacco, strong, citrus, flowers and wood. Coffee from Java has a spicy aroma while that from Wamena has a fragrant aroma and without pulp [2]. The specific flavours and aromas are attributed to the composition of commodities' metabolites. Generally, specific metabolite is responsible for particular flavours and aroma. For this reason, it is vital to recognize the characteristics of each plantation commodity based on the metabolite composition. This study investigates the origin of clove buds. This helps to maintain the flavour of a product using clove buds as a mixture. Also, the characteristics of food products can be predicted based on the origin of clove buds used due to the differences in flavour and taste between regions [3].

Metabolic profiling is a widely used approach in obtaining information related to metabolites contained in a biological sample. This is a quantitative measurement of metabolites from biological samples [4, 5]. To give meaning to the metabolites data sets, the chemo-

metrics technique was developed. This is a chemical sub-discipline that uses mathematics, statistics and formal logic to gain knowledge about chemical systems. It provides maximum relevant information by analysing metabolites data sets from biological samples [6]. Additionally, it is used in pattern recognition of metabolites data sets in complex chemical systems [3]. Pattern recognition in biological samples identifies specific metabolites or biomarkers that form a particular flavour and aroma.

Artificial neural networks have been widely used in pattern recognition [7] and other applications in various fields as shown by some researchs [8–13]. However, it has not been fully implemented, especially in clove buds. The small data sets available limit the implementation of artificial neural networks for clove buds. This is attributed to the lack of metabolite composition in the clove buds and the high cost for extracting them. Furthermore, some clove buds have zero metabolite concentration. However, this is because of inefficient tools in the laboratory to detect metabolites whose values are very small. Therefore, this study implements artificial neural networks as pattern recognition in clove buds data sets. Each origin of clove buds has specific metabolites as a biomarker.

2. MATERIALS AND METHODS

2.1. MATERIALS

This study uses clove buds data sets obtained from Kresnowati et al. [3]. It examined clove buds from four origins in Indonesia, including Java, Bali, Manado and Toli-Toli. Each origin has three regions, and

therefore, there are twelve regions in total. In the laboratory, eight experiments are carried out in each region, except for Java with only six experiments. For each experiment, 47 types of metabolites were recorded. In the matrix, data sets are 94×47 . The row and column represent the number of experiments and metabolites, respectively.

2.2. DATA PREPROCESSING

In total, the clove buds data sets have a wide range, specifically between 10^{-4} and 10. Therefore, logarithmic transformations are used to obtain reliable numerical data. Since some metabolites data have zero concentration, logarithmic transformation cannot be applied directly. This is because their concentrations range below the specified threshold. The metabolite data with zero concentration are not removed because of acting as biological markers. Therefore, they are replaced with a value of one order smaller than the smallest concentration available. In this case, the zeros are replaced with 10^{-5} . Before implementing artificial neural networks, one stage preprocessing clove buds data sets from [14] are added to normalize the values of metabolites data. The normalization ensures that each data has the same influence or contribution to determine its origin. The following normalization formula is used [15]

$$z_{kl} = \frac{x_{kl} - \bar{x}}{s}. \quad (1)$$

Here, z_{kl} is the result of normalization of x_{kl} , \bar{x} is the mean of the k -th experiment and s is

$$s = \sqrt{\sum_{k=1}^n \frac{x_{kl} - \bar{x}}{n-1}}. \quad (2)$$

2.3. ARTIFICIAL NEURAL NETWORK

Artificial neural networks are a false representation of the human brain that simulates the learning process [16]. Backpropagation and resilient propagation are learning algorithms widely used in artificial neural networks [17–27]. In this study, two different network architectures, including resilient and backpropagation, are used. The first and second architectures consist of two and one hidden layers, respectively.

2.3.1. BACKPROPAGATION LEARNING ALGORITHM

The backpropagation learning algorithm is based on the repeated use of chain rules to calculate the effect of each weight in network concerning the error function E [28].

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_i} \frac{\partial o_i}{\partial net_i} \frac{\partial net_i}{\partial w_{ij}} \quad (3)$$

where w_{ij} is the weight from j -th neuron to i -th neuron, o_i is the output, and net_i is the weighted number of neurons input i . Once the partial derivatives for each weight are known, the goal of minimizing the error function is achieved with gradient descent [28]:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \epsilon \frac{\partial E}{\partial w_{ij}}^{(t)} \quad (4)$$

where t is iteration and $0 < \epsilon < 1$ the learning rate. From the Equation (4), choosing a large learning rate (close to 1), allows for oscillations. This makes the error fall above the specified tolerance value and lessens the identification accuracy. Conversely, in case the learning rate (ϵ) is too small (close to 0), many steps are needed for convergence of the error function E . To avoid these, the backpropagation learning algorithm is expanded by adding the momentum parameter ($0 < \alpha < 1$) as shown in Equation (5). The addition of the momentum parameter also accelerates the convergence of error function [28].

$$\Delta w_{ij}^{(t+1)} = -\epsilon \frac{\partial E}{\partial w_{ij}}^{(t)} + \alpha \Delta w_{ij}^{(t-1)} \quad (5)$$

where it measures the effect of the previous step on the current one.

To activate neurons in the hidden and output layers, the sigmoid activation function is used. Three essential properties used in backpropagation and resilient propagation include bounded, monotonic and continuously differentiable. This helps to convert a weighted amount of input into an output signal for each neuron i as shown by Equation (6) [29].

$$O_i = f(I_i) = \frac{1}{1 + e^{-\sigma I_i}}. \quad (6)$$

where I_i is the input of i -th weighted number of neuron, σ the slope parameter of the sigmoid activation function and O_i the output of i -th neuron. The threshold used on the output layer for the sigmoid activation function is

$$O_i = \begin{cases} 1, & \text{if } O_i \geq 0.5 \\ 0, & \text{if } O_i < 0.5. \end{cases} \quad (7)$$

The weighted amount input is given in the following equation [29].

$$\sum_{i=1}^n w_{ij} O_i + w_{Bj} O_B. \quad (8)$$

The sum of i represents the input received from all neurons in the input layer, while B is the bias neuron. Weight w_{ij} is associated with connections from i -th neuron to j -th neuron, while w_{Bj} weight relates to the connections from the biased to j -th neuron. The weighted amount obtained in the hidden and the output layers are activated by substituting the weighted amount from Equation (8) to be an exponent in Equation (6).

2.3.2. RESILIENT PROPAGATION LEARNING ALGORITHM

Riedmiller et al. in [28] proposed a resilient propagation learning algorithm developed by the backpropagation algorithm. The algorithm directly adapts to the

weight value based on the local gradient information. Riedmiller et al. [28] introduced an update value Δ_{ij} for each weight determining the size of the weight update. The adaptive update value evolves during the learning process based on its local sight on the error function E , according to the following learning rule [28]:

$$\Delta_{ij}^t = \begin{cases} \eta^+ * \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ \eta^- * \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ \Delta_{ij}^{(t-1)}, & \text{else} \end{cases} \quad (9)$$

where $(0 < \eta^- < 1 < \eta^+)$ η^- and η^+ represents the decrease and increase factors, respectively. According to this adaptation rule, every time the partial derivative of the corresponding weight w_{ij} changes its sign, which indicates that the last update is too big and the algorithm is above the local minimum, the update value Δ_{ij} is decreased by the factor η^- . In case the derivative retains its sign, the update value slightly increases to accelerate the convergence in the shallow regions [28].

Once the update value for each weight is adjusted, the update weight itself follows the rule stating that in case the derivative is positive, the weight is decreased by its update value. If the derivative is negative, the update value is added

$$\Delta w_{ij}^t = \begin{cases} -\Delta_{ij}^{t-1}, & \text{if } \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ +\Delta_{ij}^{t-1}, & \text{if } \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ 0, & \text{else} \end{cases} \quad (10)$$

$$w_{ij}^{t+1} = w_{ij}^t + \Delta w_{ij}^t \quad (11)$$

However, in case the partial derivative sign changes, which means the previous step was too large and the minimum missed, the previous weight update is reverted:

$$\Delta w_{ij}^{(t)} = -\Delta w_{ij}^{(t-1)}, \text{ if } \frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \quad (12)$$

Due to the 'backtracking' weight step, the derivative should change its sign once again in the next step. To avoid another problem, there should be no adaptation of the update value in the succeeding step. In practice, this can be carried out by setting $\frac{\partial E}{\partial w_{ij}}^{(t-1)} = 0$ in the Δ_{ij} adaptation rule. The update values and weights are changed every time the whole set of patterns is presented to the network once (learning by epoch).

The following shows the process of adaptation and resilient propagation learning process. The **minimum(maximum)** operator is expected to provide a minimum or maximum of two numbers. The sign operator returns +1 if the argument is positive, -1 in the case it is negative, and 0 for otherwise.

For each weight and bias{

```

if(  $\frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} > 0$  ) then{
     $\Delta_{ij}^{(t)} = \text{minimum}(\Delta_{ij}^{(t-1)} * \eta^+, \Delta_{\max})$ 
     $\Delta w_{ij}^{(t)} = \text{sign}(\frac{\partial E}{\partial w_{ij}}^{(t)} * \Delta_{ij}^{(t)})$ 
     $w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$ 
}

else if(  $\frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} < 0$  ) then{
     $\Delta_{ij}^{(t)} = \text{maximum}(\Delta_{ij}^{(t-1)} * \eta^-, \Delta_{\min})$ 
     $w_{ij}^{(t+1)} = w_{ij}^{(t)} - \Delta w_{ij}^{(t-1)}$ 
     $\frac{\partial E}{\partial w_{ij}}^{(t)} = 0$ 
}

if(  $\frac{\partial E}{\partial w_{ij}}^{(t-1)} * \frac{\partial E}{\partial w_{ij}}^{(t)} = 0$  ) then{
     $\Delta w_{ij}^{(t)} = -\text{sign}(\frac{\partial E}{\partial w_{ij}}^{(t)} * \Delta_{ij}^{(t)})$ 
     $w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$ 
}
}

```

3. RESULTS AND DISCUSSIONS

In this study, the percentage of training and testing data sets are 80 % and 20 %, respectively. The metabolites data sets in matrix are 94×47 . Out of 94 rows, 75 were chosen randomly as training data sets, while the remaining were used as testing sets. The selection of training data sets is carried out randomly 30 times. Therefore, in each network architecture, there are 30 values for the percentage of identification accuracy, coefficient of determination and the mean squared error (MSE). The average is chosen as a representative of the 30 values. In each network architecture, learning rate (ϵ) 0.9, momentum parameter (α) 0.1 and maximum epoch 5000 are used with an error target of 10^{-3} . In this study, each origin is represented by a binary code. Specifically, the binary code for the Java origin is 1000, Bali 0100, Manado 0010 and Toli-Toli 0001. The calculation of the identification accuracy and MSE is shown in Equation (13) and (14).

$$\% \text{ accuracy} = \frac{a}{k} 100 \% \quad (13)$$

Where a is the number of origins identified correctly, while k is the total number. MSE calculated by the following equation [29]

$$MSE = \frac{1}{m \cdot n} \sum_{p=1}^m \sum_{k=1}^n (T_{kp} - O_{kp})^2. \quad (14)$$

where T_{kp} is the desired target, O_{kp} the network output and p the variable corresponding to the number of origins.

The suitability between the expected target and network output was evaluated based on the coefficient of determination R^2 . It was calculated using the following equation [21]

$$R^2 = 1 - \frac{\frac{1}{n} \sum_{k=1}^n (T_{kp} - O_{kp})^2}{\frac{1}{n-1} \sum_{k=1}^n (T_{kp} - \bar{T}_{kp})^2}. \quad (15)$$

Where \bar{T}_{kp} is the average desired target.

In this study, backpropagation and resilient propagation were used, each consisting of two and one hidden layers. For one hidden layer, the number of neurons was determined using the formula proposed by Shibata and Ikeda in 2009 [30], specifically $N_h = \sqrt{N_i \cdot N_o}$, where N_h , N_i , and N_o represent hidden, input and output neurons, respectively. In both the backpropagation and resilient propagation, the number of neurons used does not exceed one hidden layer. Based on Shibata and Ikeda [30] formula, the number of neurons in one hidden layer was obtained, specifically $N_h = \sqrt{N_i \cdot N_o} = \sqrt{47 \cdot 4} = 13.71$. However, in this study, it was rounded up to 15 neurons. Some experiments were conducted to evaluate the identification accuracy, and whether using one hidden layer with 15 neurons might lead to a better accuracy of identification than two hidden layers. However, the number of neurons varied, setting less than 15 neurons. For two hidden layers, experiments were conducted with the number of consecutive neurons as follows; 3-5 (8), 4-6 (10), 5-7 (12) and 6-8 (14). The number of neurons in the hidden layer never exceeded 15 neurons.

3.1. BACKPROPAGATION (B-PROP) WITH TWO HIDDEN LAYERS

In this section, the backpropagation learning algorithm with two hidden layers was used. The number of neurons in the hidden layer varied with not more than 15 neurons. There were four variations of the network architecture, including 47-3-5-4, 47-4-6-4, 47-5-7-4 and 47-6-8-4. The input layer consists of 47 neurons based on the number of metabolites. The output layer consists of 4 neurons according to the number of clove buds origins.

Table 1 shows the network architecture **47-3-5-4** gives the highest value for the identification accuracy and coefficient of determination in training and testing data sets. Similar to the *MSE*, this network architecture provides the smallest amount of both training and the testing data sets. From Table 1, increasing the number of neurons in the backpropagation with two hidden layers decreases network performance. This is in line with Shafi et al. in 2006 [31], which stated that increasing the number of neurons in the hidden layer only heightened the complexity of the network. Still, it does not increase the accuracy of the pattern recognition.

3.2. BACKPROPAGATION (B-PROP) WITH ONE HIDDEN LAYER

The backpropagation learning algorithm with one hidden layer was implemented to evaluate its result in the case of a comparison using two hidden layers. The results obtained are shown in Table 2.

Table 2 shows that the network architecture 47-15-4 identifies the clove buds origin effectively. The identification accuracy percentage is 99.91% and 99.47% for training and testing data sets, respectively. Moreover, the *MSE* value is also smaller compared to the two hidden layers.

For the backpropagation algorithm, the results show one hidden layer is better than two. This is in line with Villiers and Barnard [32], which stated that a network architecture with one hidden layer is on average better than two hidden layers. They concluded that two hidden layers are more difficult to train. Additionally, they also established that this behaviour is caused by a local minimum problem. The networks with two hidden layers are more prone to the local minimum problem during the training.

3.3. RESILIENT PROPAGATION (R-PROP) WITH TWO HIDDEN LAYERS

Resilient propagation learning algorithm contains some parameters, including the upper and lower limits as well as the decrease and increase factors. In this study, the range of update values is limited to the upper limit (Δ_{max}) = 50, the lower limit (Δ_{min}) = 10^{-6} , and the decrease and increase factors (η^-) = 0.5 and (η^+) = 1.2, respectively. The reason for choosing these values is shown in [28]. Similar to Section 3.1, the resilient propagation learning algorithm is applied to the network architecture with two hidden layers. The number of neurons vary but do not exceed 15 neurons. In this section, there are four variations of the network architecture, including 47-3-5-4, 47-4-6-4, 47-5-7-4 and 47-6-8-4.

The results in Table 3 show the network architecture **47-5-7-4** gives the highest identification accuracy of the clove buds origin. The percentage of the identification accuracy is 99.96% and 97.89% for training and testing data sets, respectively.

3.4. RESILIENT PROPAGATION (R-PROP) WITH ONE HIDDEN LAYER

In this section, the resilient propagation learning algorithm is implemented with one hidden layer. Similar to section 3.2, the number of neurons in the hidden layer is 15 neurons, and have the network architecture 47-15-4.

Table 4 shows the network architecture 47-15-4 identifies the origin of clove buds with an identification accuracy of 99.86% and 94.74% on training and testing data sets, respectively.

The network architecture of the resilient propagation algorithm, both two hidden layers and one hidden layer, provides identification results with a very high

Network Architecture	<i>MSE</i>		Accuracy (%)		R^2	
	Training	Testing	Training	Testing	Training	Testing
47-3-5-4	0.10346	0.11357	76.98	73.68	0.81	0.76
47-4-6-4	0.13084	0.13547	62	57.54	0.64	0.61
47-5-7-4	0.14889	0.15884	49.73	41.4	0.51	0.42
47-6-8-4	0.15388	0.15874	50.04	42.46	0.48	0.43

TABLE 1. Backpropagation with two hidden layers.

Network Architecture	<i>MSE</i>		Accuracy (%)		R^2	
	Training	Testing	Training	Testing	Training	Testing
47-15-4	0.0721	0.0773	99.91	99.47	0.99	0.98

TABLE 2. Backpropagation with one hidden layer.

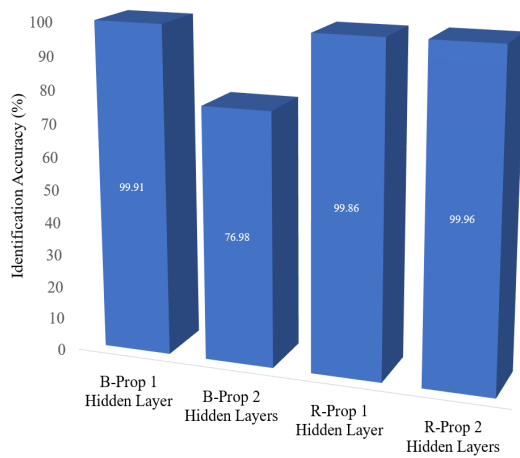


FIGURE 1. Identification accuracy percentage of training data sets.

accuracy. However, the network architecture with two hidden layers has a slightly lower accuracy.

Tables 3 and 4 show the two-layered resilient propagation with deficient neurons performs better than the single-layer having more neurons. This is in line with Santra et al. [24], which established that the performance of two hidden layers with 8-10 (18) neurons is better than that of one hidden layer with 62 neurons.

The summary of the best identification accuracy and determination coefficient are shown in Figures 1, 2, 3 and 4, respectively. For each network architecture, the smallest *MSE* in training and testing data sets are shown in Figures 5 and 6, respectively.

The results of the identification from the origins of clove buds have been obtained. In small data set categories, backpropagation with one hidden layer provides an accurate identification in the training and testing data sets. It accurately identifies the origins of clove buds obtained using the resilient propagation algorithm with two hidden layers.

The neural networks model obtained in this paper can be a reference from a scientific perspective. For instance, it can be used in future studies to identify the

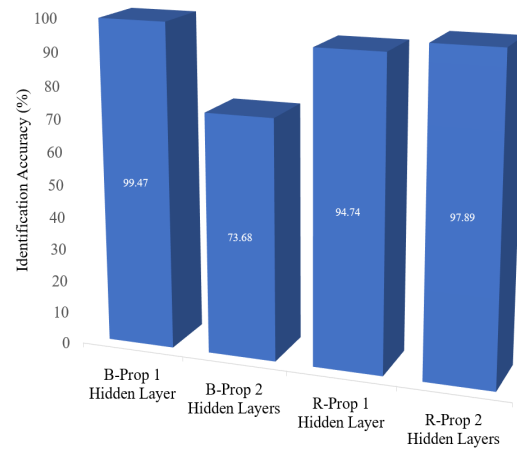


FIGURE 2. Identification accuracy percentage of testing data sets.

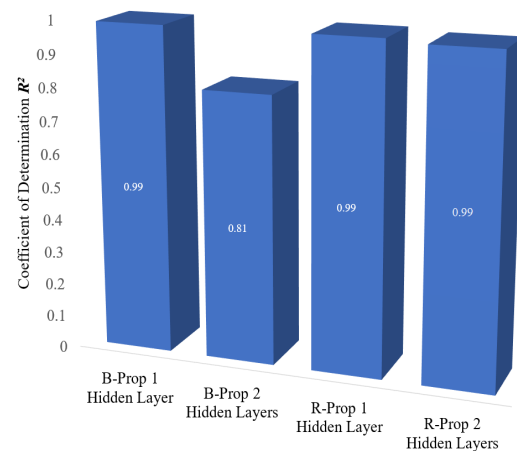


FIGURE 3. Determination coefficient of training data sets.

Network Architecture	<i>MSE</i>		Accuracy (%)		R^2	
	Training	Testing	Training	Testing	Training	Testing
47-3-5-4	0.07209	0.08111	99.73	97.37	0.98	0.95
47-4-6-4	0.07162	0.08316	99.69	96.49	0.99	0.94
47-5-7-4	0.07160	0.07961	99.96	97.89	0.99	0.96
47-6-8-4	0.07160	0.07978	99.78	97.72	0.99	0.96

TABLE 3. Resilient propagation with two hidden layers.

Network Architecture	<i>MSE</i>		Accuracy (%)		R^2	
	Training	Testing	Training	Testing	Training	Testing
47-15-4	0.07158	0.07932	99.86	94.74	0.99	0.92

TABLE 4. Resilient propagation with one hidden layer.

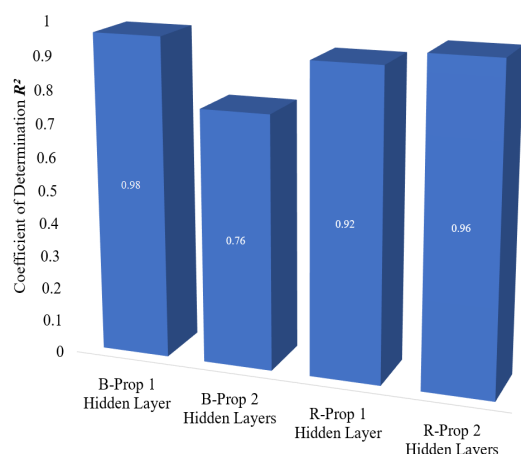
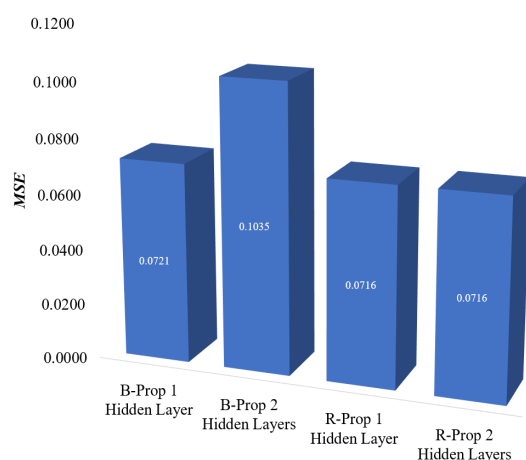
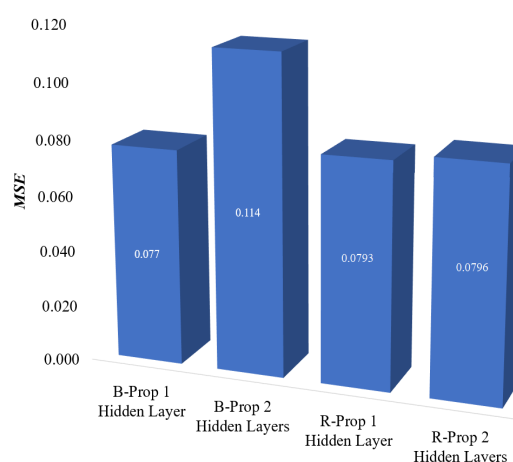


FIGURE 4. Determination coefficient of testing data sets.

FIGURE 5. *MSE* of training data sets.FIGURE 6. *MSE* of testing data sets.

origin of various plantation commodities with small metabolites data sets. At the moment, the most appropriate way of determining the origin of a plantation commodity is qualitative, relying on the services of flavourist to evaluate the flavour and taste. This is because each commodity has a specific flavour and taste based on the origin of its region. Furthermore, the different origins of clove buds data sets have not been reported in the literature and thus no direct comparison can be presented in this paper.

4. CONCLUSIONS

This paper demonstrated the potential and ability of a neural network approach with backpropagation and resilient propagation learning algorithms. It was meant to identify the clove buds origin based on metabolites composition. The work was divided into two parts, the first one being an identification of the clove buds origin using the backpropagation learning algorithm. In this algorithm, two network architectures were constructed. One having a single hidden layer and the second one having two. The results showed that the use of one hidden layer gives the clove buds origin identification accurately, specifically 99.91 %

and 99.47 % in training and testing data sets, respectively. The second step involved the identification of the clove buds origin using a resilient propagation learning algorithm. In this algorithm, two network architectures were constructed. One having a single hidden layer and the second one having two. The results showed that the use of two hidden layers gives an accurate clove buds origin identification, including 99.96 % and 97.89 % in training and testing data sets, respectively. From these results, it was concluded that for an identification of small metabolites data sets from a plantation commodity, the backpropagation algorithm with one hidden layer and the resilient propagation algorithm with two hidden layers should be used. This paper also confirmed the contribution of artificial neural networks to the pattern recognition of metabolites data sets obtained by the metabolic profiling technique.

ACKNOWLEDGEMENTS

The authors express gratitude to the government of Indonesia, especially Endowment Fund for Education (LPDP - Lembaga Pengelola Dana Pendidikan), for its funding.

REFERENCES

- [1] L. Broto. *Derivatisasi minyak cengkeh, dalam cengkeh: Sejarah, Budidaya dan Industri*, chap. 23. Indesco Jakarta dan Biologi UKSW Salatiga, 2014 (in Indonesian).
- [2] Coffeeland 2010. Jenis kopi Arabika terbaik dari berbagai daerah di Indonesia (in Indonesian). <https://coffeeland.co.id/product-category/kopi-supply/>.
- [3] M. T. A. P. Kresnowati, R. Purwadi, M. Zunita, et al. Metabolite profiling of four origins indonesian clove buds using multivariate analysis. *Report Research Collaboration PT HM Sampoerna Tbk and Institut Teknologi Bandung (confidential report)* 2018.
- [4] J. Kopka, A. Fernie, W. Weckwerth, et al. Metabolite profiling in plant biology: platforms and destinations. *Genome biology* **5**(6):109, 2004. DOI:10.1186/gb-2004-5-6-109.
- [5] S. P. Putri, E. Fukusaki. *Mass spectrometry-based metabolomics: a practical guide*. CRC Press, 2016. DOI:10.1007/s13361-015-1246-3.
- [6] D. L. Massart, B. G. M. Vandeginste, S. N. Deming, et al. *Chemometrics: a textbook*. Elsevier Amsterdam, 1988. DOI:10.1002/cem.1180020409.
- [7] C. T. Leondes. *Image Processing and Pattern Recognition*. Elsevier, 1998.
- [8] B. Samir, A. Boukelif. New approach for online arabic manuscript recognition by deep belief network. *Acta Polytechnica* **58**(5), 2018. DOI:10.14311/AP.2018.58.0297.
- [9] A. N. Ponce, A. A. Behar, A. O. Hernández, V. R. Sitar. Neural networks for self-tuning control systems. *Acta Polytechnica* **44**(1), 2004.
- [10] M. Chvalina. Demand modelling in telecommunications: Comparison of standard statistical methods and approaches based upon artificial intelligence methods including neural networks. *Acta Polytechnica* **49**(2):48–52, 2009.
- [11] I. Bukovsky, M. Kolovratnik. A neural network model for predicting nox at the melnik 1 coal-powder power plant. *Acta Polytechnica* **52**(3):17–22, 2012.
- [12] P. Kutilek, S. Viteckova. Prediction of lower extremity movement by cyclograms. *Acta Polytechnica* **52**(1), 2012.
- [13] D. Novák, D. Lehký. Neural network based identification of material model parameters to capture experimental load-deflection curve. *Acta Polytechnica* **44**(5-6), 2004.
- [14] Rustam, A. Y. Gunawan, M. T. A. P. Kresnowati. The hard c-means algorithm for clustering indonesian plantation commodity based on metabolites composition. In *Journal of Physics: Conference Series*, vol. 1315, p. 012085. IOP Publishing, 2019. DOI:10.1088/1742-6596/1315/1/012085.
- [15] T. Beltramo, M. Klocke, B. Hitzmann. Prediction of the biogas production using ga and aco input features selection method for ann model. *Information Processing in Agriculture* 2019. DOI:10.1016/j.inpa.2019.01.002.
- [16] L. Fausett. *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall, Inc., 1994.
- [17] I. Aizenberg, C. Moraga. Multilayer feedforward neural network based on multi-valued neurons (mlmvn) and a backpropagation learning algorithm. *Soft Computing* **11**(2):169–183, 2007. DOI:10.1007/s00500-006-0075-5.
- [18] E. M. Johansson, F. U. Dowla, D. M. Goodman. Backpropagation learning for multilayer feed-forward neural networks using the conjugate gradient method. *International Journal of Neural Systems* **2**(04):291–301, 1991. DOI:10.1142/S0129065791000261.
- [19] T. T. Pleune, O. K. Chopra. Using artificial neural networks to predict the fatigue life of carbon and low-alloy steels. *Nuclear Engineering and Design* **197**(1-2):1–12, 2000. DOI:10.1016/S0029-5493(99)00252-6.
- [20] K. L. Kaiser, S. P. Niculescu, G. Schüürmann. Feed forward backpropagation neural networks and their use in predicting the acute toxicity of chemicals to the fathead minnow. *Water Quality Research Journal* **32**(3):637–658, 1997. DOI:10.2166/wqrj.1997.037.
- [21] E. El Tabach, L. Adishirinli, N. Gascoin, G. Fau. Prediction of transient chemistry effect during fuel pyrolysis on the pressure drop through porous material using artificial neural networks. *Journal of analytical and applied pyrolysis* **115**:143–148, 2015. DOI:10.1016/j.jaap.2015.07.010Get.
- [22] R. Chayjan, M. ESNA-ASHARI. Isosteric heat and entropy modeling of pistachio cultivars using neural network approach. *Journal of Food Processing and Preservation* **35**(4):524–532, 2011. DOI:10.1111/j.1745-4549.2010.00498.x.
- [23] A. D. Anastasiadis, G. D. Magoulas, M. N. Vrahatis. New globally convergent training scheme based on the resilient propagation algorithm. *Neurocomputing* **64**:253–270, 2005. DOI:10.1016/j.neucom.2004.11.016.

- [24] A. K. Santra, N. Chakraborty, S. Sen. Prediction of heat transfer due to presence of copper–water nanofluid using resilient-propagation neural network. *International Journal of Thermal Sciences* **48**(7):1311–1318, 2009. DOI:10.1016/j.ijthermalsci.2008.11.009.
- [25] L. M. Patnaik, K. Rajan. Target detection through image processing and resilient propagation algorithms. *Neurocomputing* **35**(1-4):123–135, 2000. DOI:10.1016/S0925-2312(00)00301-5.
- [26] D. Fisch, B. Sick. Training of radial basis function classifiers with resilient propagation and variational bayesian inference. In *2009 International Joint Conference on Neural Networks*, pp. 838–847. IEEE, 2009. DOI:10.1109/IJCNN.2009.5178699.
- [27] M. Shiblee, B. Chandra, P. K. Kalra. Learning of geometric mean neuron model using resilient propagation algorithm. *Expert Systems with Applications* **37**(12):7449–7455, 2010. DOI:10.1016/j.eswa.2010.04.018.
- [28] M. Riedmiller, H. Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Proceedings of the IEEE international conference on neural networks*, vol. 1993, pp. 586–591. San Francisco, 1993. DOI:10.1109/ICNN.1993.298623.
- [29] P. Bhagat. *Pattern recognition in industry*. Elsevier, 2005. DOI:10.1016/B978-0-08-044538-0.X5054-X.
- [30] K. Shibata, Y. Ikeda. Effect of number of hidden neurons on learning in large-scale layered neural networks. In *2009 ICCAS-SICE*, pp. 5008–5013. IEEE, 2009.
- [31] I. Shafi, J. Ahmad, S. I. Shah, F. M. Kashif. Impact of varying neurons and hidden layers in neural network architecture for a time frequency application. In *2006 IEEE International Multitopic Conference*, pp. 188–193. IEEE, 2006. DOI:10.1109/INMIC.2006.358160.
- [32] J. De Villiers, E. Barnard. Backpropagation neural nets with one and two hidden layers. *IEEE transactions on neural networks* **4**(1):136–141, 1993. DOI:10.1109/72.182704.